Assignment 5-5 Armstrong Numbers

An Armstrong number is a positive integer that is the sum of its own digits each raised to the power of the number of digits. For example, 153 is an Armstrong number because $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$. Likewise, 1634 is an Armstrong number because $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 256 = 1634$.

In this problem you have to determine whether a given positive integer is an Armstrong number or not.

Input

The input consists of t ($30 \le t \le 40$) test cases. The first line of the input contains only positive integer t. Then t test cases follow. Each test case consists of exactly one line with a positive integer n which is less than 2^{31}.

Output

For each line of input, there will be one line of output. If n is an Armstrong number print 'Armstrong', otherwise print 'Not Armstrong' (without the quotes).

Sample Input

2
153
154

Sample Output

Armstrong
Not Armstrong

Requirements

You are required to write two recursive functions `int sumPowerDigits( int n )` and `int power( int d, int p )` to complete the following program which solves this problem. The function `int sumPowerDigits( int n )` returns the sum of the k^{th} powers of all digits of `n`, where k is the number of digits of `n`; and the function `int power( int d, int p )` returns the `p`-th power of `d`.

```
#include<iostream>
using namespace std;

// returns the sum of the k-th powers of digits of n,
// where k is the number of digits of n.
int sumPowerDigits( int n );
```

```
// returns the p-th power of d
int power( int d, int p );

int numDigits; // the number of digits of n

int main()
{
   int numCases;
   cin >> numCases;
   for( int i = 1; i <= numCases; i++ )
   {
      int n;
      cin >> n;

      numDigits = 0; // the number of digits of n
      for( int i = n; i > 0; i /= 10 )
         numDigits++;

      if( n == sumPowerDigits( n ) )
         cout << "Armstrong" << endl;
      else
         cout << "Not Armstrong" << endl;
   }
}

int sumPowerDigits( int n )
{



}

// returns the p-th power of d
int power( int d, int p )
{



}
```